cellebrite
delivering mobile expertise

What Happens When You Press that Button?

Explaining Cellebrite UFED Data Extraction Processes

Table of Contents

UFED Basics

Cellebrite makes mobile device evidence extraction available on two different platforms: the UFED Touch, or the UFED 4PC. The UFED 4PC is extraction software that can be installed on any PC platform, accessible and securable in the same way as any other PC-based software.

The UFED Touch consists of standalone proprietary hardware, with the UFED software installed on the Microsoft® Windows® Embedded Standard 2009 platform. Users can only access limited functionality—shut down, log on/off—and cannot access the Windows operating system.

The UFED Touch and UFED 4PC interfaces and architecture are exactly the same. UFED software is designed to execute only read commands, and to prevent the opportunity to alter it to issue write commands to mobile devices. While the operator should document which platform, version, and extraction type were used, no other differences fundamentally exist between UFED Touch and UFED 4PC extractions.

UFED operators should also adhere to the same best practices for both Touch and 4PC platforms that they do for any forensic computer installation:

- Isolate the forensic machine from the Internet while performing forensic examinations.
- Don't store digital evidence on any computer that is or will be connected to the Internet at any time.
- If performing an over-the-air software update, the operator should not simultaneously have an evidence device or evidence storage connected to the forensic machine.
- Extract mobile device evidence to a storage medium specifically designed and prepared for that purpose: a Flash drive, external hard drive, a location on an internal forensic network, or internal drive or partition within the forensic computer.

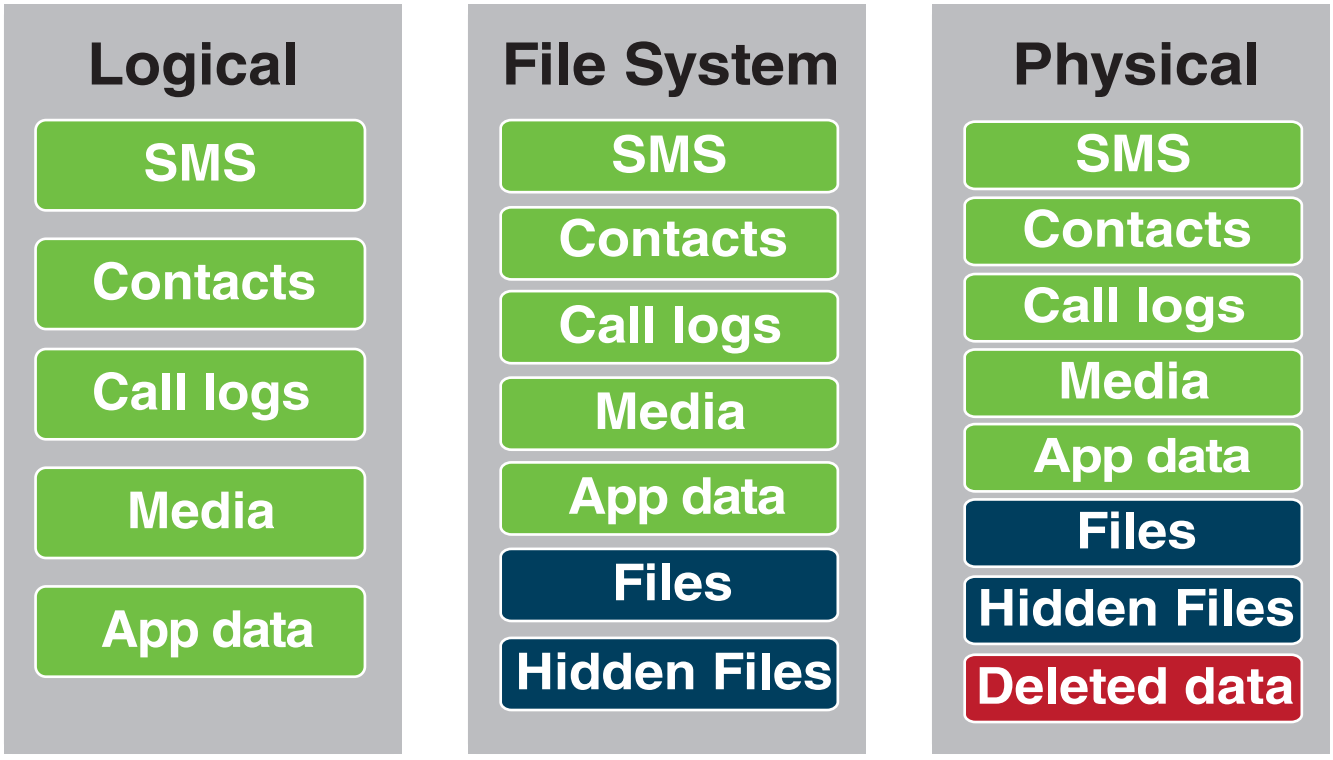
As of April 2014, a permission management feature was included in the UFED 3.0 update. UFED Permission Management is an optional feature that allows forensic lab administrators to enforce operator accountability and limit search scopes, either based on extraction type or to specific data types.

This way, administrators can control who performs extractions based on their level of training, job responsibilities, or other "right to know, need to know" criteria as specified in their organization's policies or standard operating procedures. Cellebrite encourages all customers who distribute mobile evidence collection in their organizations to use UFED Permission Management.

Extraction Types

There are two different methods of mobile device data extraction: logical and physical. (A third extraction type, the file system extraction, technically falls under the "logical" heading.) Different data types, if they are supported for the device, are available from each extraction category, as shown in the graphic below.

In the rare instances when the extraction fails, the user must simply start the extraction over. The failure does not affect the quality or integrity of evidentiary data because it only affects the transmission of data from the device, not the data on the device.



In most cases, mobile phones are connected to the UFED device via a USB cable connection, which communicates with the phone to extract its data. The use of a USB connection provides a proven reliable channel upon which to copy data from evidence device to the forensic image.

Depending on the subject phone's OS, logical extractions may instead use USB/Bluetooth protocol APIs or, with older devices, serial protocols in order to extract the data. Operators should document which connection type they used for each extraction.

Logical extraction

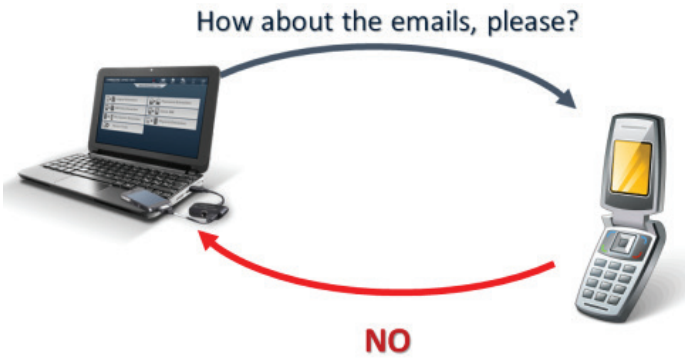
Logical extraction of data is performed, for the most part, through a designated API (Application Programming Interface), available from the device vendor. Just as the API allows commercial third-party apps to communicate with the device OS (operating system), it also enables forensically sound data extraction.

Upon connection, the UFED loads the relevant vendor API to the device. The UFED then makes read-only API calls to request data from the phone. The phone replies to valid API requests to extract designated content items such as text messages (SMS), phonebook entries, pictures, etc.

From a technical standpoint, API-based logical extraction is straightforward to implement and the results are provided in a readable format. However, the logical method is limited to the scope of content the specific vendor has made available through its API.

Pictures taken via third-party app, for example, are likely stored in a folder that is different from the default.

Therefore, the API will not see that they exist and will not make them available to a UFED logical extraction. To access this data, an examiner would need to access the file system and examine the data associated with the particular application in question. In addition, not all devices have a common interface to extract emails, and the API will not be applicable.



Logical extractions of iOS devices

In July 2011 Cellebrite identified the need for a faster means of extracting data from iOS devices. The pre-UFED Touch hardware, the UFED Classic or UFED 36, could take many hours to perform these extractions. Cellebrite solved the problem by implementing iOS extraction within its analysis software, UFED Physical Analyzer, as of version 2.1.

It is possible for iOS device extractions to differ between the UFED Touch/UFED 4PC interface and the UFED Physical Analyzer.

That's because the UFED Touch/UFED 4PC obtains the Apple iTunes backup interface using its API, the Apple File Connection (AFC)—the same interface used to back up the device to a computer.

File system extraction with UFED Physical Analyzer is almost identical to physical extraction in that it relies on a boot loader to access the device's memory; however, rather than obtain a bit-for-bit image including unallocated space, the software extracts only the device file system. This extraction process is proprietary rather than dependent on Apple's API.

Moreover, UFED Physical Analyzer makes three different types of iTunes backup ("Advanced Logical") extractions possible.

- **Method 1** like the UFED Touch, relies on the iTunes backup using Apple's backup infrastructure
- **Method 2** extracts backup data if the device is encrypted and the UFED operator does not know the device passcode
- **Method 3** is recommended for both encrypted and unencrypted jailbroken devices

How does the examiner know which method to choose?

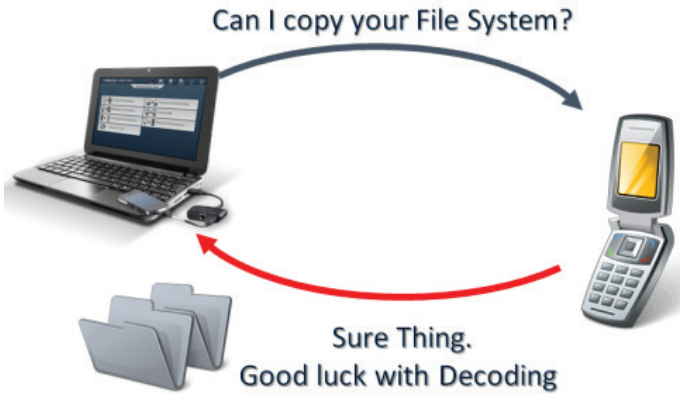
The UFED Physical Analyzer interface automatically selects the appropriate extraction method—based on the device's backup configuration, jailbreak status, model, and iOS version—but the operator has the option to use other methods as well, and to combine the data sets. The interface explains which data is available with each extraction method. Users should document which method(s) they used and why they used it, when possible.

File system extraction

Another logical method extends the examiner's reach to the phone's live partition. Available with the UFED Ultimate license, a file system extraction uses different device-specific methods to copy the file system. While these are comparable to the API used in logical methods, they use different sets of built-in protocols, depending on the OS. The mix of protocols often differs from device family to device family.

In some cases, not only with iOS devices as described above but also with Android and BlackBerry® models, it may be necessary to rely on device backup files to make available files, hidden files, and other data that is not necessarily accessible through the phone's API.

This can include some user deleted and hidden data contained within SQLite databases, including web history, email headers, EXIF data on images, and system data.



Decoding

The decoding process translates the raw data within a database file to a recognizable format. Data extracted via APIs and backups require no decoding because it is intrinsic to these methods, which present media files such as pictures and videos as they are seen on the device.

However, data within other database files, such as those that contain text messages, must be separately decoded to parse out the messages. UFED Physical Analyzer automatically performs this decoding process, presenting decoded data both in human-readable format, and as raw data as stored in the device's memory.

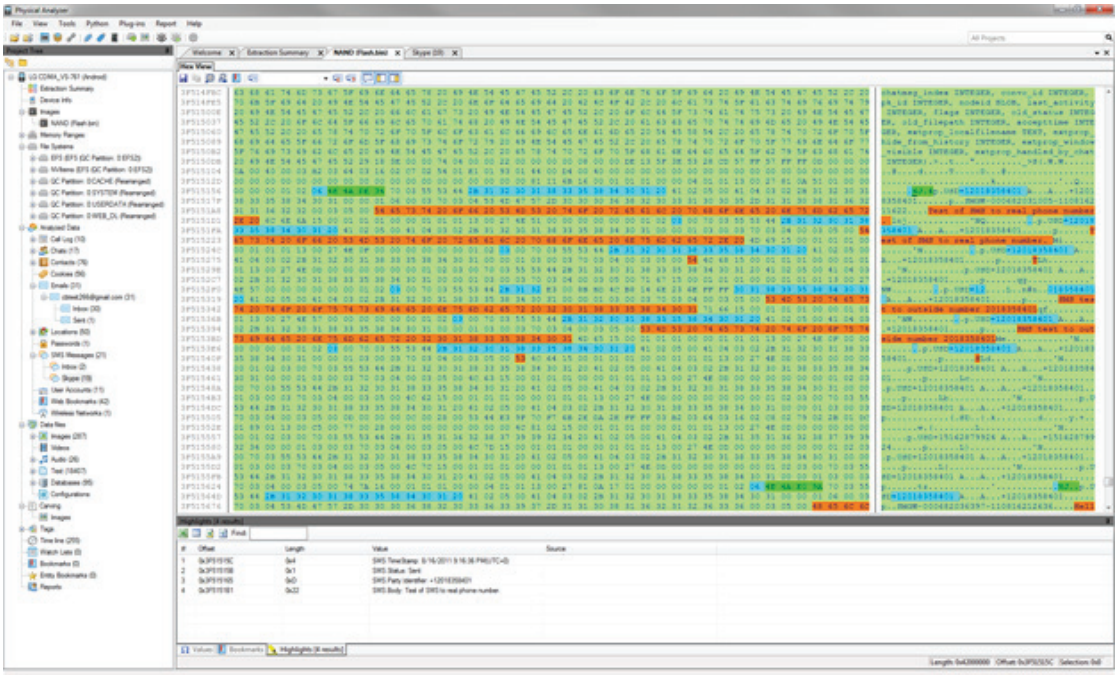


Figure 1: UFED Physical Analyzer displays data in human-readable format, as well as in its raw state as stored on the device.

Can decoding miss some data?

It is possible for automatic decoding to miss some data. Decoding relies on programming that tells the software to look for and interpret data in certain places on the device, based on patterns from previous make, model and operating system versions. Cellebrite's access to mobile device manufacturers and carriers makes this easier for the UFED to accomplish than it does for other tools, but it is not a guarantee.

This is particularly an issue when it comes to app data, which is stored within SQLite database files and plist files on iOS devices. UFED Physical Analyzer identifies that these files exist, and certain database file extractions from some Android and BlackBerry® smartphone apps — including Facebook, Skype, Twitter, Viber, Yahoo messenger, Whatsapp, TigerText and others—are even possible through UFED Logical.

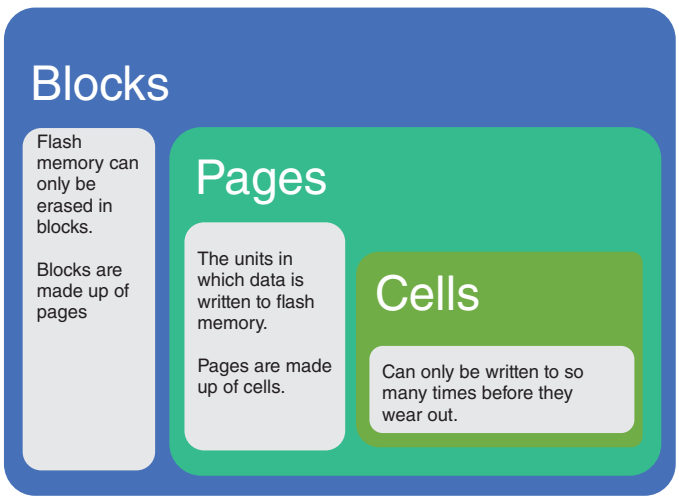
This data also requires decoding through the process described above. However, it is unfeasible, due to the large number of apps available, for UFED Physical Analyzer software to be programmed to parse every SQLite database file that is present. If a database exists that UFED Physical Analyzer knows to look for—if it supports decoding for a particular app—it will decode and present the data. However, an obscure or unsupported app will not show up in the "analyzed items" section of the Physical Analyzer software.

Wear leveling and garbage collection

Unlike traditional hard disk drives, which simply overwrite unneeded data blocks with new data, mobile devices' flash memory must erase unneeded data blocks before new data can be written.

This process extends the memory life, storing data more efficiently by distributing it (and thus, write/erase cycles) evenly across each solid state drive (SSD) chip. Known as wear leveling, this process complicates the extraction and decoding process because it is possible for data to look different from one extraction to the next.

This is because of "garbage collection," the process of erasing the unneeded data blocks. Blocks are composed of pages, where data is written, but the data can only be erased in blocks. When pages within a block become unnecessary, wear leveling rewrites the "good" pages into an empty block, and garbage collection erases the unneeded blocks.



Garbage collection happens in the background, but its speed and frequency can vary from SSD to SSD. As a result, it may be that the examiner performs an extraction before garbage collection has occurred. If this happens, deleted data may be written multiple times in multiple locations on the SSD.
(See Figure 2.)

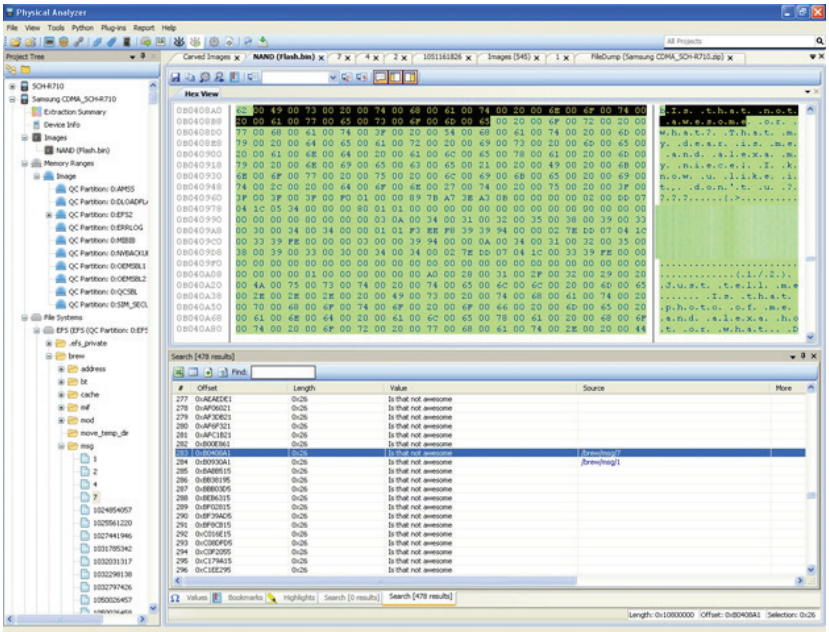


Figure 1: wear leveling in action. the same text message shows up multiple times because garbage collection has not yet happened

While this can be beneficial in terms of recovering deleted data, it can also complicate forensic exams—not only by increasing the amount of data to be parsed, but also by potentially changing the pattern on which the decoding process relies to parse data. It is possible for wear leveling to affect the offset (found in the hex code) so that it doesn't match the pattern in the decoding program.

Whether due to a lack of decoding or wear leveling, the data is likely present in the extraction, but forensic examiners should be prepared to use the hexadecimal viewer within UFED Physical Analyzer to carve for additional data if needed. Additionally, they should be prepared to explain why the tool extracted but did not decode the data, and if possible, how they validated that the carved data was stored on the device.

Physical extraction

To allow the most comprehensive and detailed analysis of the device, Cellebrite's physical extraction capability accesses the additional data layers, in both allocated and unallocated space, that construct the phone's physical memory. These layers include three different groups of content pertinent to investigators:

1. "Logical" content unavailable through API (e.g. call logs on smartphones and feature phones)

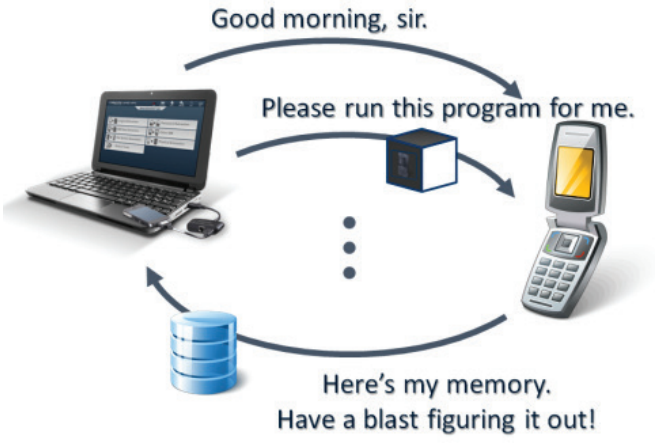
2. Deleted content

3. Content that the phone collects without any user action (and sometimes without user knowledge). For example: wi-fi networks, GPS locations, web history, email headers and EXIF data on images, and system data.

The physical extraction allows the examiner to access this data by creating a bit-for-bit copy of the mobile device's flash memory. As with the file system extraction, the data within this copy can be decoded via UFED Physical Analyzer. Seeing where the data is located within the device's memory enables the analyst to interpret the data.

Boot loaders

A common method used to physically extract binary files from mobile phones is through "rescue mode" or "download mode". Operating in this mode, mobile phones are designed to allow the insertion of a small piece of code, called boot loaders, into the RAM during start-up.



In the commercial world, this allows the operator to use a product called a "flasher box" to insert the boot loader and overwrite the device's flash memory, so as to upgrade the device or change service providers.

Although flasher boxes have been successfully used in a forensic context, they were not developed to be forensic tools. They are not "read-only" devices, and as a result they make it possible for unskilled users to make inadvertent changes to the evidence.

Some mobile forensic products incorporate these third-party boot loaders. However, this is a "black box" solution because there is no access to the device's source code.

The utilization of third-party boot loaders may involve risks of modifying the evidence and in some cases even cause phone malfunction.

Why Cellebrite boot loaders are forensically sound

While Cellebrite relies on the boot loader concept, its boot loaders are designed in-house around each individual device platform with its variety of chipsets, peripherals, memory chip interfaces, and USB/serial controllers to be efficient and deliver quick, accurate results using a repeatable, reproducible methodology. By controlling every part of the code running on the device, Cellebrite ensures that the process is non-intrusive and that nothing in the device's user partition is changed.

It also avoids data integrity concerns associated with jailbreaking an iOS device, rooting an Android, or other methods that bypass a smartphone's factory settings, including built-in security and other restrictions, to provide administrative "root" access to its operating system.

During the initial stage of the device's booting, the UFED sends the boot loader to the device's RAM memory. The device will start running the boot loader, but will not continue its regular booting procedure into the OS. The Cellebrite boot loaders then execute "read only" actions that extract evidence from mobile devices and leave no artifacts behind.

Each boot loader is specifically designed to read the contents of the device's memory, and send it back to the UFED.

In the majority of devices, Cellebrite's proprietary boot loader can bypass all security mechanisms, even if the device is locked, without jailbreaking, rooting or flashing the device. Because the boot loader contains nothing but code used to read the various memory chips on the device, and does not write to the memory chips on the device data at any stage during or after the extraction process, the data extraction and passcode bypass processes are forensically sound.

Other physical extraction methodologies

Even so, newer devices, including some Android smartphones, don't have a built-in functionality to upload boot loaders.
In some cases this may necessitate "temporary rooting" in order to gain access to the information. During this process, Cellebrite clients are uploaded to the device to enable temporary rooting and thus, extraction. Following the extraction, the client is uninstalled and the device boots as usual, non-rooted.

In other instances, UFED users have the option to use a different type of client. This can leave a footprint in the user data partition, notably the potential for writing to small, unallocated areas of the storage medium. The client is installed in the next available bit of unallocated space, which then becomes allocated for that purpose.

This type of installation is comparable to walking into a snowy crime scene to retrieve a murder weapon. The investigator may leave his or her own footprints behind, but this necessity is acceptable in court as long as it is carefully documented. As a result, the UFED prompts users by first asking if they want to install a client.

In addition, the UFED has a setting that by default uninstalls the client after it is written to the device.

While this is useful for intelligence professionals who must operate covertly, law enforcement who use the client should follow their department's protocol about whether or not to uninstall the client, or should document its use and whether or not they uninstalled it.

Some agencies, for example, may require examiners to always disable the "automatic uninstall" setting, declare and document its use and leave the client in place. Other agencies may require this action only for suspect phones, but allow the client to be uninstalled from a victim's phone as long as its use is documented.

Likewise, another family of several LG phones (a very small percentage) require, when the device is locked and there is no other way to access the data, flashing of a proprietary boot loader to the phone.

This necessitates rewriting the phone's memory, permanently changing the device boot loader to Cellebrite's own.

The UFED warns the user when this is about to happen, and still does not change any user data. The chance of damaging the device in this case is very low; there is a small chance of overwriting data in unallocated space. Examiners should document the situations in which this is necessary, and why.

Authentication and reporting

Once logical, file system, and physical extractions are complete, the UFED generates an extraction file, along with a .UFD (text) file that tells UFED Physical Analyzer what the extraction is. The .UFD file contains information about the extraction, such as which UFED was used (including its serial number); start time, finish time, and date; and hash information. With iOS physical extractions, the .UFD file also contains decryption keys. For binary images, it may contain some information to ease the decoding procedure.

For logical extractions, the .UFD file references a .ZIP or backup file; for physical extractions, the .UFD file references an .IMG (image) file or a .BIN (binary) image file, depending on how the extraction was performed and on which platform.

Any data which the UFED extracts is hashed using SHA256 and/or MD5 algorithms, which helps to maintain data authenticity. These algorithms are included within the .UFD file.

However, UFED Physical Analyzer does not create a forensic "container" comparable to an EnCase .E01 file.

As a last step, UFED Physical Analyzer creates a report. Report formats can vary, with logical, file system or physical extractions reported in the UFED report package (UFDR), Microsoft Word® or Excel®, Open Document Format (ODF), HTML, PDF, or XML files.

When necessary, some of these formats can be imported into other forensic and data management tools for additional analysis.

For attorneys and other authorized personnel, Cellebrite makes available a free application, UFED Reader, available in the installation package with each UFED license. UFED Reader allows anyone to view, search and filter results from the .UFDR report package. Interested attorneys should ask licensed users to download and send a copy of UFED Reader.

In Conclusion

The extraction processes employed by Cellebrite UFED can seem complex, and it is wise to ensure that the investigators or examiners being called as witnesses have a good enough grasp of the technology to explain it in a way that a jury can understand.

To this end, Cellebrite strongly encourages all users to attend certification training in order to best understand—and explain—how to extract, decode, analyze and document mobile device evidence using these advanced methodologies.

Certification training is available worldwide and in multiple delivery modes, including online and in-class. To learn more, visit: **http://www.cellebritelearningcenter.com.**

Glossary of Terms

ADB: Android Debugging Bridge. A command line, client/server tool that allows developers to communicate with an Android device. ADB can be used to install and uninstall apps, run shell commands, backup and restore a device, and so on. In a mobile forensics context it can be important, in some makes and models, to enabling physical and file system extractions from Android devices.

Allocated space: The area on a device's memory that stores data in an organized manner, and contains its operating system and user data. Logical extractions obtain data from allocated space only.

API: Application Programming Interface. Specifies how apps and firmware on a mobile device should interact with one another.

Boot loader: A small piece of code that is inserted into the RAM during start-up. In the commercial wireless world, this allows flashing of firmware. In the forensic world, it allows a non-intrusive means of accessing and copying user data into a forensic image.

Carving: The process of finding data contained within the hexadecimal code, apart from what the forensic software has automatically offered. Carving can become necessary when the forensic tool parses data from unsupported apps, with deleted data including images, and other situations with file system and physical extractions.

Client/agent: A client is used during logical extractions. It is a very small application that is temporarily installed on a limited number of Android, older Windows Mobile, Palm OS, and Symbian models. The client is unlike a boot loader in that, rather than be installed to the device RAM, it acts like any other third-party app by installing to the device ROM. It does not overwrite any data; it will not install, for example, on a device whose memory is full. It provides enough access to the device's file system that allows UFED to index the file system and determine how many files exist, then extract the data. It is automatically removed from the device after the extraction is complete. Users are encouraged to document when the UFED prompts them to use the client, and whether they proceed with the use.

Decoding: The process of translating raw hexadecimal data into an easily readable format. An automatic process within UFED Physical Analyzer and UFED Logical Analyzer, decoding renders data easier for the examiner to find and analyze. From file system and physical extractions, the examiner always has the option to examine hexadecimal code within the raw data.

Extraction: The process of obtaining mobile device data and storing it in an approved location for processing.

Jailbreaking/rooting: A jailbroken iOS device or a rooted Android device is one whose owner has taken steps to bypass its factory settings, including built-in security and other restrictions. Jailbreaking an iOS device allows the user to install third-party apps from sources other than the App Store, while rooting an Android device provides administrative "root" access to its operating system. UFED solutions do not rely on jailbreaking or permanent rooting to perform forensic extractions, as other mobile forensic tools do.

SQLite database: A database file format often used for data storage. Commonly used for storage of mobile and application data, but many smartphones may use .db files, .plists, and other file formats as well.

Unallocated space: The area on a device's memory outside the defined file system that is available to write data to. Very often, deleted data or fragments can be found and carved from unallocated space.